

10-20-00

EK538216787US

A

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

10/19/00



Docket No. AUS9-2000-0370-US1

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

Transmitted herewith for filing is the patent application of Inventor(s):

William Hsiao-Yu Ku,
Joey Allen Perry and
John Shih-Yuan Wang,For: MONITORING MODIFICATIONS TO ENVIRONMENT VARIABLES


Enclosed are also:

- ☒ 12 Pages of Specification including an Abstract
☒ 9 Pages of Claims
☒ 4 Sheet(s) of Drawings
☒ A Declaration and Power of Attorney
☒ Form PTO 1595 and assignment of the invention to IBM Corporation

CLAIMS AS FILED

FOR	Number Filed		Number Extra		Rate		Basic Fee (\$710)
Total Claims	32	-20 =	12	X	\$ 18	=	\$ 216.00
Independent Claims	14	-3 =	11	X	\$ 80	=	\$ 880.00
Multiple Dependent Claims	0			X	\$260	=	\$ 00.00
Total Filing Fee							= \$1,806.00

- ☒ Please charge \$1,806.00 to IBM Corporation, Deposit Account No. 09-0447.
☒ The Commissioner is hereby authorized to charge payment of the following fees associated with the communication or credit any over payment to IBM Corporation, Deposit Account No. 09-0447. A duplicate copy of this sheet is enclosed.
☒ Any additional filing fees required under 37CFR § 1.16.
☒ Any patent application processing fees under 37CFR § 1.17.

Respectfully,

Marilyn Smith Dawkins
Reg. No. 31,140
Intellectual Property Law Dept.
IBM Corporation
11400 Burnet Road 4054
Austin, Texas 75758
Telephone: (512) 823-0094

09690457-101900

Docket No. AUS9-2000-0370-US1

MONITORING MODIFICATIONS TO ENVIRONMENT VARIABLES

BACKGROUND OF THE INVENTION

5

1. Technical Field:

The present invention relates generally to the field of computer software and, more particularly, to monitoring changes to environment variables.

10

2. Description of Related Art:

Computer use has increased exponentially during the past several years. Much of this growth has been due to the increasing use of personal computers for home use due to recent sharp decreases in the price of computers as technology advances. This increase in the number of computers in use has also been spurred by the recent explosion of the Internet.

Thus large numbers of people with little or no computer expertise are interacting with computers on a daily basis. Novice users are purchasing and loading software applications onto their computers from a variety of sources without regard for what other software applications may exist on their computer and without regard as to how the different software applications will integrate with each other. Many of these software applications include and use some of the same executable files as other software applications already loaded onto the user's computer. When a new software application is loaded, the user may end up having multiple copies of the same executable file stored in different locations in the user's computer. However, the two copies of the

DOCKET "AUS9-2000-0370-US1"

Docket No. AUS9-2000-0370-US1

executable file may be different versions.

Thus, if one software application attempts to run the wrong version of the executable file, problems may occur. The problem of duplicate files is not limited to
5 situations arising from inexperienced computer users as discussed above. Duplicate files also may pose a problem to even more sophisticated computer users. For example, a user may, for various reasons, expressly desire to have multiple versions of a software application or data file
10 available on the computer. However, ensuring that the proper file is selected is still a problem.

One reason some software errors occur due to the existence of duplicate files is that the incorrect one is often selected due to the order of the directories in an
15 environment variable. For example, assume that the PATH environment variable is defined as "PATH=C:\x\bin;C:\y\bin" and a.exe exists in both C:\x\bin and C:\y\bin. When the user executes a.exe, the one in the C:\x\bin directory will be used. In some
20 cases, this is exactly what the users desires. However, in other cases, the user wishes to execute C:\y\bin\a.exe, but the user is unaware that a.exe also exists in C:\x\bin.

Thus, duplicate files can cause numerous problems
25 and often these problems are very difficult to debug. Therefore, it would be desirable to have a method, system, and apparatus for managing the path sequence of environment variables to prevent the existence of duplicate path sequences in an environment variable.

Docket No. AUS9-2000-0370-US1

SUMMARY OF THE INVENTION

5 The present invention provides a method, system, and
program for automatically invoking an environment
variable manager whenever a path sequence for an
environment variable may be modified. The environment
variable manager then corrects the path sequence of the
10 environment variable in a data processing system. In one
embodiment, an environment variable manager monitors the
data processing system for any change effecting any of
the environment variables within the data processing
system. If a change effecting the environment variable
15 is detected, the environment variable manager modifies
the environment variable to ensure that a proper file is
found and used when the file is selected by a user or
requested by a running application program. Therefore,
when duplicate files exist on the data processing system,
20 the environment variable manager ensures that the
incorrect file is not used when the file is requested by
a user or requested by a running application program.

006707 25406969

Docket No. AUS9-2000-0370-US1

BRIEF DESCRIPTION OF THE DRAWINGS

5 The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself, however, as well as a preferred mode of use, further objectives and advantages thereof, will best be understood by reference to the following detailed
10 description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

Figure 1 depicts a block diagram of a data processing system in which the present invention may be implemented;

15 **Figure 2** depicts a block diagram illustrating a path management system in accordance with the present invention;

Figure 3 depicts a process flow and program function for updating the path sequence of an environment variable
20 when a directory is manually deleted in accordance with the present invention; and

Figure 4 depicts a process flow and program function for removing duplicate file names from a path sequence of an environment variable in accordance with the present
25 invention.

006707-101900

Docket No. AUS9-2000-0370-US1

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

With reference now to the figures, and in particular
5 with reference to **Figure 1**, a block diagram of a data
processing system in which the present invention may be
implemented is illustrated. Data processing system **100**
employs a peripheral component interconnect (PCI) local
bus architecture. Although the depicted example employs
10 a PCI bus, other bus architectures, such as Micro Channel
and ISA, may be used. Processor **102** and main memory **104**
are connected to PCI local bus **106** through PCI bridge
108. PCI bridge **108** may also include an integrated
memory controller and cache memory for processor **102**.
15 Additional connections to PCI local bus **106** may be made
through direct component interconnection or through
add-in boards.

In the depicted example, local area network (LAN)
adapter **110**, SCSI host bus adapter **112**, and expansion bus
20 interface **114** are connected to PCI local bus **106** by
direct component connection. In contrast, audio adapter
116, graphics adapter **118**, and audio/video adapter (A/V)
119 are connected to PCI local bus **106** by add-in boards
inserted into expansion slots. Expansion bus interface
25 **114** provides a connection for a keyboard and mouse
adapter **120**, modem **122**, and additional memory **124**. In
the depicted example, SCSI host bus adapter **112** provides
a connection for hard disk drive **126**, tape drive **128**,
CD-ROM drive **130**, and digital video disc read only memory
30 drive (DVD-ROM) **132**. Typical PCI local bus

Docket No. AUS9-2000-0370-US1

implementations will support three or four PCI expansion slots or add-in connectors.

An operating system runs on processor **102** and is used to coordinate and provide control of various components within data processing system **100** in **Figure 1**. The operating system may be a commercially available operating system, such as OS/2, which is available from International Business Machines Corporation. "OS/2" is a trademark of International Business Machines Corporation. An object oriented programming system, such as Java, may run in conjunction with the operating system, providing calls to the operating system from Java programs or applications executing on data processing system **100**. Instructions for the operating system, the object-oriented operating system, and applications or programs are located on a storage device, such as hard disk drive **126**, and may be loaded into main memory **104** for execution by processor **102**.

Those of ordinary skill in the art will appreciate that the hardware in **Figure 1** may vary depending on the implementation. For example, other peripheral devices, such as optical disk drives and the like, may be used in addition to or in place of the hardware depicted in **Figure 1**. The depicted example is not meant to imply architectural limitations with respect to the present invention. For example, the processes of the present invention may be applied to multiprocessor data processing systems.

With reference now to **Figure 2**, a block diagram illustrating a path management system is depicted in accordance with the present invention. System **208** may be

Docket No. AUS9-2000-0370-US1

implemented as, for example, data processing system 100
in **Figure 1**. Duplicate files on the same system 208 will
cause problems when both files are in an environment
variable's 204 path sequence. An environment variable is
5 an item of data that is updated by the operating system,
Web server or other control program. Environment
variables typically reside in memory, such as, for
example, memory 124 in **Figure 1**, and can be read by
applications to determine the current status of the
10 system 208. Environment variables contain data such as
time, date, path sequence, version number, login
information and so on. One example of an environment
variable is the PATH environment variable. Other
examples of environment variables, as will be recognized
15 by one of ordinary skill in the art, include CLASS PATH,
LOC PATH, and LIB PATH.

When a path sequence is modified or when duplicate
files are created or installed in the system 208,
environment variable manager 202 informs a user of this
20 modification through I/O device interface 206. I/O
device interface 206 may comprise a plurality of
interfaces and/or devices and provides an interface to
numerous devices such as, for example, a keyboard and/or
mouse for receiving user input and, for example, a video
25 display terminal for displaying information to a user.
Environment variable manager 208 then prompts the user,
through I/O device interface 206 for actions to be taken
to correct the problem.

When a directory is manually deleted from system
30 208, some path sequences of environment variables 204
which contain that directory may not be affected, but the

0060457-101900

Docket No. AUS9-2000-0370-US1

non-existent directory may cause confusion at a later time. Therefore, environment variable manager 202 informs the user at that moment so that the non-existent directory may be deleted from the path sequence of the affected environment variables 204.

With reference now to **Figure 3**, a process flow and program function for updating the path sequence of an environment variable when a directory is manually deleted is depicted in accordance with the present invention.

Once an environment variable manager, such as, for example, environment variable manager 202 in **Figure 2**, detects the deletion of a directory (step 302) from the system, such as, for example, system 208 in **Figure 2**, the environment variable manager presents a message to the user that a directory has been deleted and prompts the user for an appropriate action (step 304). The user may select to allow the environment variable manager to automatically update the affected environment variables, such as, for example, environment variables 204 in **Figure 2**, or may, alternatively, choose to modify the affected environment variables manually.

Thus, the environment variable manager determines from the user input whether the user has selected an automatic or manual update to the environment variables (step 306). If the user selects an automatic update, the environment variable manager searches and finds all references to the deleted directory in the environment variables (step 308). Once the affected environment variables have been found, the environment variable manager deletes all references to the deleted directory from the affected environment variables (step 310). If

Docket No. AUS9-2000-0370-US1

the user selects a manual update, the environment variable manager searches and finds all references to the deleted directory in environment variables (step **312**) and presents the list of all affected environment variables to the user (step **314**). The user may then manually edit each affected environment variable to correct the problem.

Returning now to **Figure 2**, when a software product is installed on system **208**, additional directories may be added to the path sequence of some environment variables **204**. This could result in duplicate files existing in system **208** and environment variable manager **202** informs the user, through I/O device interface **206** such that the problem may be corrected. Furthermore, when an environment variable **204** is modified manually or by the system, this also could result in duplicate files existing in the path sequence of that particular environment variable **204**. Since the first path found in the environment variable **204** will be the one selected, problems may arise if the undesired one is selected first. Therefore, environment variable manager **202** monitors and detects modification of environment variables **204** and determines whether duplicate path sequences exist. If duplicate files exist in the path sequence of one or more of environment variables **204**, environment variable manager **202** prompts the user via I/O device interface **206** for the appropriate action and then corrects the problem.

With reference now to **Figure 4**, a process flow and program function for removing duplicate file names from a path sequence of an environment variable is depicted in

Docket No. AUS9-2000-0370-US1

accordance with the present invention. The environment variable manager, such as, for example, environment variable manager **202** in **Figure 2**, monitors environment variables, such as, for example environment variables **204** in **Figure 2**. If the environment variable manager detects that an environment variable has been modified (step **402**), environment variable manager determines whether duplicate files exist in the path sequence of that environment variable (step **404**). If no duplicate files exist in the path sequence of the modified environment variable, then no further action is taken.

If, however, duplicate files do exist in the path sequence of the modified environment variable, the environment variable manager prompts the user to select the appropriate file name that is the correct file (step **406**). Once the environment variable manager receives the selection of the correct file from the user (step **408**), the environment variable manager then removes the incorrect file or files from the path sequence of the modified environment variable (step **410**). Thus, the path sequence of the environment variable is corrected to ensure that the proper file is used when necessary.

It is important to note that while the present invention has been described in the context of a fully functioning data processing system, those of ordinary skill in the art will appreciate that the processes and program function of the present invention are capable of being distributed in the form of a computer readable medium of instructions in a variety of forms and that the present invention applies equally regardless of the particular type of signal bearing media actually used to

Docket No. AUS9-2000-0370-US1

carry out the distribution. Examples of computer readable media include recordable-type media such a floppy disc, a hard disk drive, a RAM, and CD-ROMs and transmission-type media such as digital and analog
5 communications links.

The description of the present invention has been presented for purposes of illustration and description, but is not intended to be exhaustive or limited to the invention in the form disclosed. Many modifications and
10 variations will be apparent to those of ordinary skill in the art. The embodiment was chosen and described in order to best explain the principles of the invention, the practical application, and to enable others of ordinary skill in the art to understand the invention for
15 various embodiments with various modifications as are suited to the particular use contemplated.

006707 " 101900

Docket No. AUS9-2000-0370-US1

CLAIMS:

What is claimed is:

- 5 1. A method for correcting a path sequence of an environment variable in a data processing system, the method comprising:
- monitoring the data processing system for a change effecting the environment variable; and
- 10 responsive to detection of the change effecting the environment variable, altering the environment variable to ensure that a proper file is found and used when selected by one of a user and a running application program.
- 15 2. The method as recited in claim 1, wherein the change is a change effecting the path sequence of the environment variable.
- 20 3. The method as recited in claim 1, wherein the change is a deletion of a directory from the data processing system.
4. The method as recited in claim 1, wherein the change
- 25 is the installation of a software application onto the data processing system.
5. The method as recited in claim 1, wherein the change is the modification of the contents of the environment
- 30 variable.
6. The method as recited in claim 1, wherein the step

006707 45406960

Docket No. AUS9-2000-0370-US1

of altering the environment variable comprises removing references to all but one of duplicate files in the path sequence of the environment variable.

5 7. The method as recited in claim 6, wherein the all but one duplicate file is selected by a user.

8. A method for modifying at least one of a plurality of environmental variables to ensure a more efficient
10 operation of a data processing system; the method comprising:

detecting the deletion of a directory from the data processing system;

determining whether any of the plurality of
15 environment variables contain a reference to the directory; and

responsive to a determination that at least one of the plurality of environment variables contains a reference to the directory, removing the reference to the
20 directory in effected ones of the plurality of environment variables.

9. A method for correcting modifications to an environment variable in a data processing system, the
25 method comprising:

detecting that an environment variable has been modified;

responsive to a determination that duplicate files exist in a path sequence of the environment variable,
30 prompting a user to select the correct one of the duplicate files; and

removing all incorrect ones of the duplicate files

0060457 10100
DOCKET 45400960

Docket No. AUS9-2000-0370-US1

from the path sequence of the environment variable.

10. A computer program product in a computer readable
media for use in a data processing system for correcting
5 a path sequence of an environment variable in the data
processing system, the computer program product
comprising:

first instructions for monitoring the data
processing system for a change effecting the environment
10 variable; and

second instructions, responsive to detection of the
change effecting the environment variable, for altering
the environment variable to ensure that a proper file is
found and used when selected by one of a user and a
15 running application program.

11. The computer program product as recited in claim 10,
wherein the change is a change effecting the path
sequence of the environment variable.

12. The computer program product as recited in claim 10,
wherein the change is a deletion of a directory from the
data processing system.

13. The computer program product as recited in claim 10,
wherein the change is the installation of a software
application onto the data processing system.

14. The computer program product as recited in claim 10,
30 wherein the change is the modification of the contents of
the environment variable.

00690457 101900

Docket No. AUS9-2000-0370-US1

15. The computer program product as recited in claim 10, wherein the step of altering the environment variable comprises removing references to all but one of duplicate files in the path sequence of the environment variable.

5

16. The computer program product as recited in claim 15, wherein the all but one duplicate file is selected by a user.

10 17. A system for correcting a path sequence of an environment variable in a data processing system, the system comprising:

first means for monitoring the data processing system for a change effecting the environment variable;

15 and

second means, responsive to detection of the change effecting the environment variable, for altering the environment variable to ensure that a proper file is found and used when selected by one of a user and a running application program.

20

18. The system as recited in claim 17, wherein the change is a change effecting the path sequence of the environment variable.

25

19. The system as recited in claim 17, wherein the change is a deletion of a directory from the data processing system.

30 20. The system as recited in claim 17, wherein the change is the installation of a software application onto the data processing system.

00690457 101900

Docket No. AUS9-2000-0370-US1

21. The system as recited in claim 17, wherein the change is the modification of the contents of the environment variable.

5

22. The system as recited in claim 17, wherein the step of altering the environment variable comprises removing references to all but one of duplicate files in the path sequence of the environment variable.

10

23. The system as recited in claim 22, wherein the all but one duplicate file is selected by a user.

15

24. A method for managing environment variables in a data processing system, comprising:

20

automatically invoking an environment variable manager whenever at least one of the following events occur a) a directory is deleted; b) a product is installed on the data processing system; c) a product is uninstalled on the data processing system; and d) a path sequence of a given environment variable is manually modified by a user;

25

determining, by the environment variable manager, if the occurring event causes a modification to an affected path sequence of any environment variable; and

30

enabling at least one of a) a correction to the affected path sequence and b) a display to the user of an interface for informing the user about a need for a correction of the affected path sequence.

25. A method for managing environment variables in a data processing system, comprising:

Docket No. AUS9-2000-0370-US1

automatically invoking an environment variable manager whenever a path sequence for an environment variable is modified;

determining, by the environment variable manager, if
5 duplicate files exist in the path sequence of the environment variable; and

enabling a display of each environment variable determined to have duplicate files in the path sequence to a user for correction.

10

26. A method for managing environment variables in a data processing system, comprising:

determining if a directory, within a path sequence of any environment variable, is manually deleted by a
15 user; and

enabling at least one of the following a) an automatic deletion of the directory from the path sequence of the environment variable; and b) a display of an interface to inform the user to delete the directory
20 from the path sequence of any affected environment variables.

27. A data processing system having means for managing environment variables, comprising:

25 means for automatically invoking an environment variable manager whenever at least one of the following events occur a) a directory is deleted; b) a product is installed on the data processing system; c) a product is uninstalled on the data processing system; and d) a path
30 sequence of a given environment variable is manually modified by a user;

means for determining, by the environment variable

DOCKET # AUS9-2000-0370-US1

Docket No. AUS9-2000-0370-US1

manager, if the occurring event causes a modification to an affected path sequence of any environment variable; and

means for enabling at least one of a) a correction
5 to the affected path sequence and b) a display to the user of an interface for informing the user about a need for a correction of the affected path sequence.

28. A data processing system having means for managing
10 environment variables, comprising:

means for automatically invoking an environment variable manager whenever a path sequence for an environment variable is modified;

means for determining, by the environment variable
15 manager, if duplicate files exist in the path sequence of the environment variable; and

means for enabling a display of each environment variable determined to have duplicate files in the path sequence to a user for correction.
20

29. A data processing system having means for managing environment variables in a data processing system, comprising:

means for determining if a directory, within a path
25 sequence of any environment variable is manually deleted by a user; and

means for enabling at least one of the following a) an automatic deletion of the directory from the path sequence of the environment variable; and b) a display of
30 an interface to inform the user to delete the directory from the path sequence of any affected environment variables.

00690457 101900

Docket No. AUS9-2000-0370-US1

30. A computer program on a computer readable medium having program code means for managing environment variables in a data processing system, comprising:

5 instruction means for automatically invoking an environment variable manager whenever at least one of the following events occur a) a directory is deleted; b) a product is installed on the data processing system; c) a product is uninstalled on the data processing system; and
10 d) a path sequence of a given environment variable is manually modified by a user;

instruction means for determining, by the environment variable manager, if the occurring event causes a modification to an affected path sequence of any
15 environment variable; and

instruction means for enabling at least one of a) a correction to the affected path sequence and b) a display to the user of an interface for informing the user about a need for correction of the affected path sequence.
20

31. A computer program on a computer readable medium having program code means for managing environment variables in a data processing system, comprising:

instruction means for automatically invoking an
25 environment variable manager whenever a path sequence for an environment variable is modified;

instruction means for determining, by the environment variable manager, if duplicate files exist in the path sequence of the environment variable; and
30

instruction means for enabling a display of each environment variable determined to have duplicate files in the path sequence to a user for correction.

09690457 101900

Docket No. AUS9-2000-0370-US1

32. A computer program on a computer readable medium having program code means for managing environment variables in a data processing system, comprising:

5 instruction means for determining if a directory, within a path sequence of any environment variable, is manually deleted by a user; and

instruction means for enabling at least one of the following a) an automatic deletion of the directory from
10 the path sequence of the environment variable; and b) a display of an interface to inform the user to delete the directory from the path sequence of any affected environment variables.

00690457 101900

Docket No. AUS9-2000-0370-US1

ABSTRACT OF THE DISCLOSURE

MONITORING MODIFICATIONS TO ENVIRONMENT VARIABLES

5

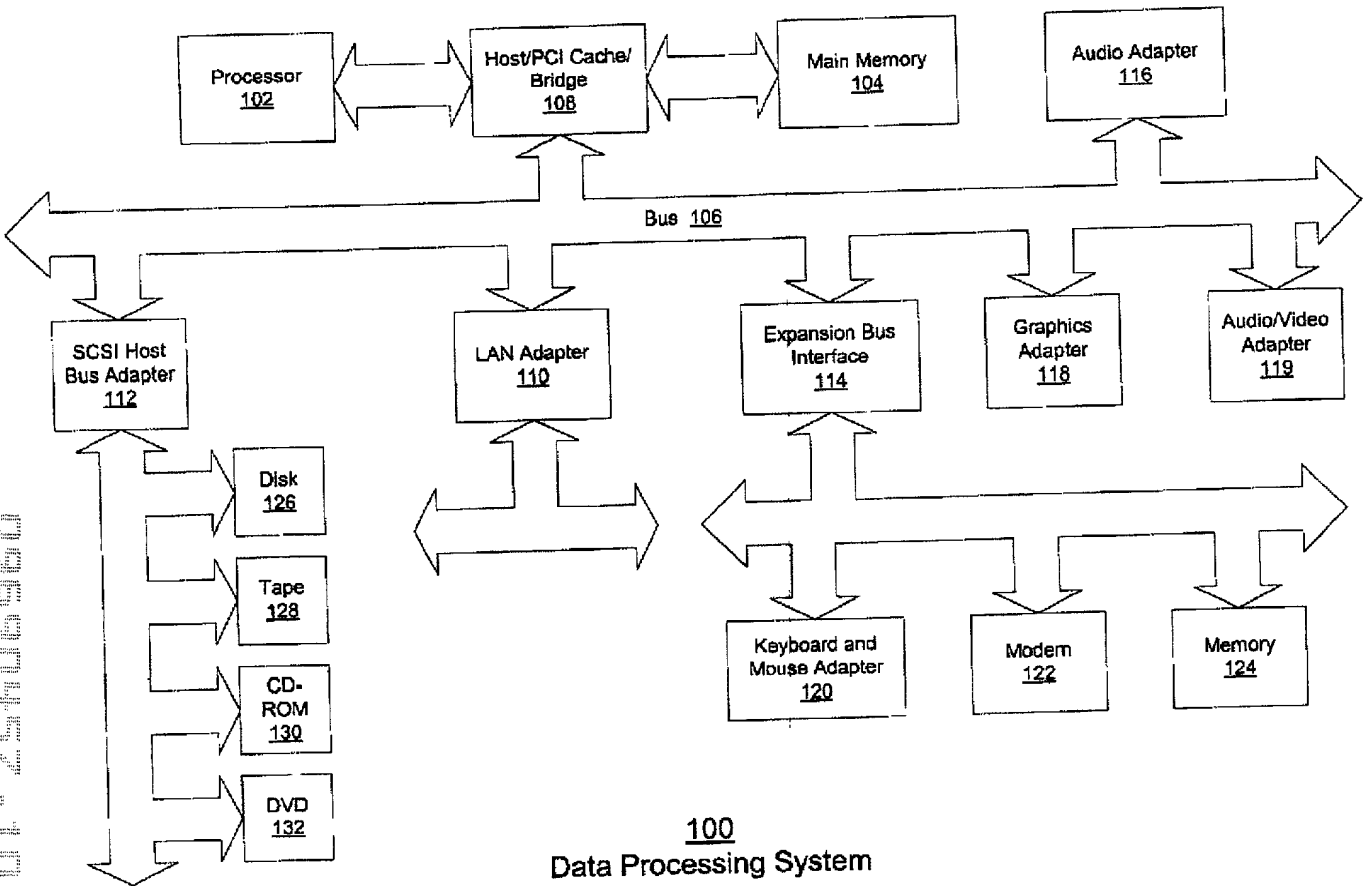
A method, system, and apparatus for correcting a path sequence of an environment variable in a data processing system is provided. In one embodiment, an environment variable manager monitors the data processing system for any change effecting any of the environment variables within the data processing system. If a change effecting the environment variable is detected, the environment variable manager modifies the environment variable to ensure that a proper file is found and used when the file is selected by a user or a running application program. Therefore, when duplicate files exist on the data processing system, the environment variable manager ensures that the incorrect file is not used when the file is requested by a user or a running application program.

10

15

20

00690457-101900



100
Data Processing System

Figure 1

AUS9-2000-0370-US1

184

006707 25406360

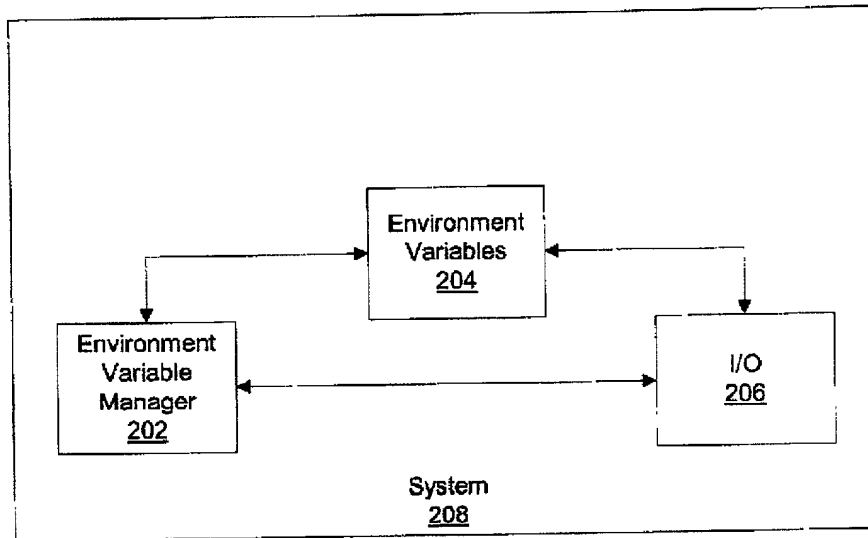


Figure 2

AUS9-2000-0370-US1

284

```
graph TD
    Start([Begin]) --> 302[Detect deletion of a directory.  
302]
    302 --> 304[Present message to user that directory  
has been deleted and prompt for  
action.  
304]
    304 --> 306{Has  
user selected  
automatic or manual update to  
environment variables?  
306}
    306 -- Automatic --> 308[Search and find all references to  
deleted directory in environment  
variables.  
308]
    306 -- Manual --> 312[Search and find all references to  
deleted directory in environment  
variables.  
312]
    308 --> 310[Delete references to deleted directory  
in environment variables.  
310]
    312 --> 314[Present list of all affected environment  
variables.  
314]
    310 --> End([End])
    314 --> End
```

The flowchart, labeled 300, describes a process for detecting and deleting references to deleted directories in environment variables. It begins with a 'Begin' terminal, leading to step 302: 'Detect deletion of a directory.' This is followed by step 304: 'Present message to user that directory has been deleted and prompt for action.' A decision diamond at step 306 asks: 'Has user selected automatic or manual update to environment variables?'. If the user selects 'Automatic', the process proceeds to step 308: 'Search and find all references to deleted directory in environment variables.', then to step 310: 'Delete references to deleted directory in environment variables.', and finally to the 'End' terminal. If the user selects 'Manual', the process proceeds to step 312: 'Search and find all references to deleted directory in environment variables.', then to step 314: 'Present list of all affected environment variables.', and finally to the 'End' terminal.

AUS9-2000-0370-US1

3884

006T07" 23406960

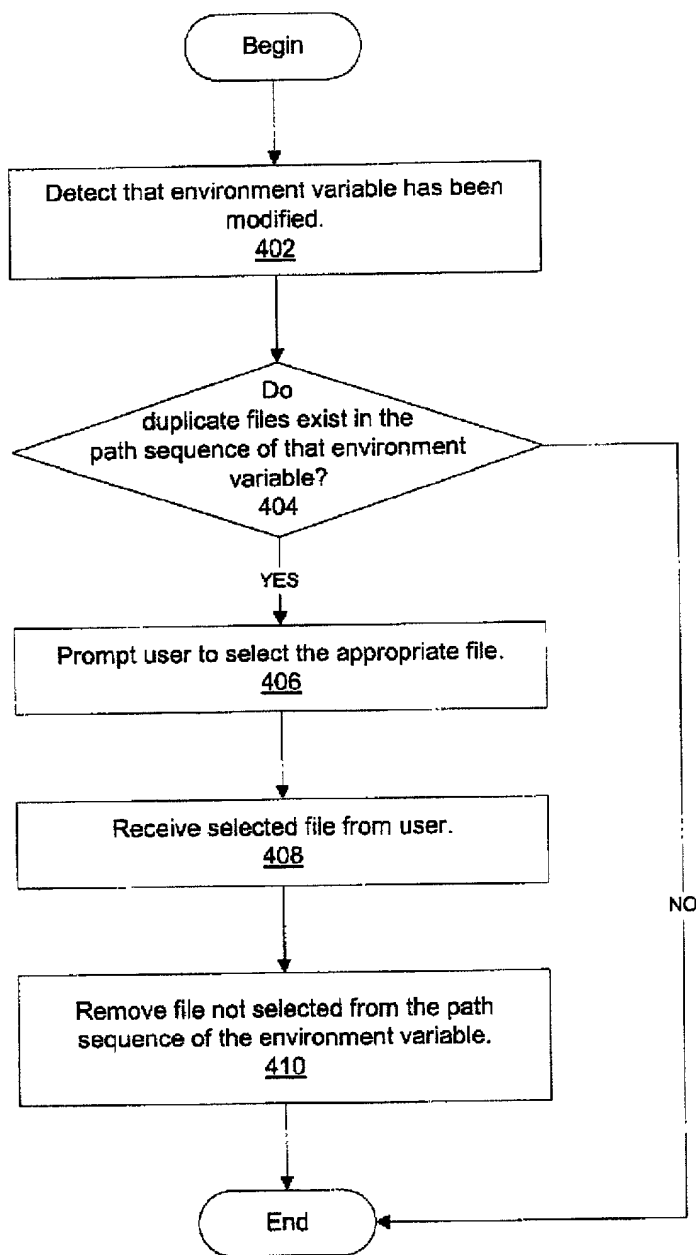


Figure 4

AUS9-2000-0370-US1

484

**DECLARATION AND POWER OF ATTORNEY FOR
PATENT APPLICATION**

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

MONITORING MODIFICATIONS TO ENVIRONMENT VARIABLES

the specification of which (check one)

X is attached hereto.

_____ was filed on _____
as Application Serial No. _____
and was amended on _____
(if applicable)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, §1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s):	Priority Claimed	
_____	_____	_____ Yes _____ No
(Number)	(Country)	(Day/Month/Year)

I hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose information material to the patentability of this application as defined in Title 37, Code of Federal Regulations, §1.56 which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

(Application Serial #)

(Filing Date)

(Status)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorneys and/or agents to prosecute this application and transact all business in the Patent and Trademark Office connected therewith.

John W. Henderson, Jr., Reg. No. 26,907; Thomas E. Tyson, Reg. No. 28,543; James H. Barksdale, Jr., Reg. No. 24,091; Casimer K. Salys, Reg. No. 28,900; Robert M. Carwell, Reg. No. 28,499; Douglas H. Lefevre, Reg. No. 26,193; Jeffrey S. LaBaw, Reg. No. 31,633; David A. Mims, Jr., Reg. 32,708; Volel Emile, Reg. No. 39,969; Anthony V. England, Reg. No. 35,129; Leslie A. Van Leeuwen, Reg. No. 42,196; Christopher A. Hughes, Reg. No. 26,914; Edward A. Pennington, Reg. No. 32,588; John E. Hoel, Reg. No. 26,279; Joseph C. Redmond, Jr., Reg. No. 18,753; Marilyn S. Dawkins, Reg. No. 31,140; Mark E. McBurney, Reg. No. 33,114; Duke W. Yee, Reg. No. 34,285; Colin P. Cahoon, Reg. No. 38,836; Stephen R. Loe, Reg. No. 43,757; Stephen J. Walder, Jr., Reg. No. 41,534; Charles D. Stepps, Jr., Reg. No. 45,880; and Stephen R. Tkacs, Reg. No. P-46,430.

Send correspondence to: Duke W. Yee, Carstens, Yee & Cahoon, LLP, P.O. Box 802334, Dallas, Texas 75380 and direct all telephone calls to Duke W. Yee, (972) 367-2001

FULL NAME OF SOLE OR FIRST INVENTOR: William Hsiao-Yu Ku

INVENTORS SIGNATURE: Hsiao-Yu Ku DATE: Oct. 16, 2000

RESIDENCE: 6506 Sans Souci Cove
Austin, Texas 78759

CITIZENSHIP: United States

POST OFFICE ADDRESS: SAME AS ABOVE

FULL NAME OF SECOND INVENTOR: Joey Allen Perry

INVENTORS SIGNATURE: Joey Allen Perry DATE: 10-17-2000

RESIDENCE: 1032 Disraeli Circle
Pflugerville, Texas 78660

CITIZENSHIP: United States

POST OFFICE ADDRESS: SAME AS ABOVE

DOCKET # 25406960

DOCKET NUMBER: AUS9-2000-0370-US1

FULL NAME OF THIRD INVENTOR: John Shih-Yuan Wang

INVENTORS SIGNATURE: John Shih-Yuan Wang

DATE: 10/16/2000

RESIDENCE: 1435 Circle Ridge Drive
Austin, Texas 78746

CITIZENSHIP: United States

POST OFFICE ADDRESS: SAME AS ABOVE

006707 43406960